

ALIVE: An Agent-based Framework for Dynamic and Robust Service-oriented Applications

J. Vázquez-Salceda
Univ. Politècnica de Catalunya
jvazquez@lsi.upc.edu

F. Dignum
Universiteit Utrecht
dignum@cs.uu.nl

W. W. Vasconcelos
University of Aberdeen
w.w.vasconcelos@abdn.ac.uk

S. Clarke
Trinity College Dublin
Siobhan.Clarke@cs.tcd.ie

J. Padget
University of Bath
jap@cs.bath.ac.uk

M. Palau Roig
Tech Media Telecom Factory
manel.palau@tmfactory.com

ABSTRACT

Service-oriented systems are becoming more and more nodes in a digital, dynamic ecosystem requiring the identification and establishment of flexible, spontaneous collaboration activities between services. This requires new engineering approaches that can integrate new functionalities and behaviours into existing running systems of active, distributed and interdependent processes. Here, we present the approach of the FP7 ALIVE¹ project to the use of formal models of *coordination* and *organisation* mechanisms to deliver a flexible, high-level means to describe the structure of interactions between services in the environment. We propose to build on the current activities in service-oriented engineering by defining three levels: (i) An organisational level models the organisational structure of executing and interlinked services and the context around them. (ii) A coordination level provides flexible ways to model interaction between the services. (iii) These two levels connect with existing (semantic) Web services, which contain semantic descriptions to make components aware of their social context and of the rules of engagement with other services.

Categories and Subject Descriptors

I.2.11 [Computing Methodologies]: Artificial Intelligence—*Distributed Artificial Intelligence, Multiagent systems*

General Terms

Design, Reliability

Keywords

Normative Systems, Social and Organisational Structure, Environments, Organisations and Institutions

1. Introduction

New generations of networked applications based on the notion of software services which can be dynamically deployed, adjusted and composed will make it possible to create radically new types of software systems. These systems shall be able to communicate and

¹This work has been performed in the framework of the FP7 project ALIVE ICT- 215890, which is funded by the European Community (<http://www.ist-alive.eu>)

Cite as: ALIVE: An Agent-based Framework for Dynamic and Robust Service-oriented Applications, J. Vázquez-Salceda, W. W. Vasconcelos, J. Padget, F. Dignum, S. Clarke, M. Palau Roig et al., *Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, van der Hoek, Kaminka, Lespérance, Luck and Sen (eds.), May, 10–14, 2010, Toronto, Canada, pp. 1637-1638
Copyright © 2010, International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

reconfigure flexibly at runtime, adapt to their environment and dynamically combine sets of building block services into new applications. In order to achieve this objective, the ALIVE framework combines model driven development with coordination and organisational mechanisms, providing support for live (i.e., highly dynamic) and open systems of services. The framework adopts the latest Semantic Web technologies to connect to existing service-oriented systems. We will demonstrate the implemented framework which validates and tests the ALIVE approach in three use cases from three industrial partners respectively: dynamic crisis management, communication in entertainment domains and dynamic orchestration of distributed services on interactive community displays.

2. ALIVE Framework

The ALIVE framework aims to support the design, deployment and maintenance of distributed systems by (1) allowing the coordination, reorganisation and adaptation of Web services, (2) following operational constraints defined in the organisation level; and (3) adapting to the dynamic nature of Web Services at runtime. The framework is supported by a set of tools, all integrated as plug-ins in a Eclipse development environment.

2.1 Organisation Level

The organisational model views an organisation as a social system, and describes what the aims and the concerns of the organisation are with respect to the social system. The OperettA tool is an organisational modelling tool. Its function is to create and manage the organisational model of a given distributed system. The designer is able to design the whole organisational level of a given distributed system through abstract concepts such as objectives, roles, obligations, violations, sanctions and high-level interaction diagrams that only identify critical states (called landmarks) and landmark patterns. The organisational model is specified in terms of four structures: (1) The *social structure* specifies objectives of the society, its roles, role dependencies and what kind of model governs coordination (see Fig. 1 (B)). (2) The *interaction structure* describes interaction moments, as scene scripts, representing a society task that requires the coordinated action of several roles, and gives a partial ordering of scene scripts, which specify the intended interactions between roles (see Fig. 1 (A)). (3) The *normative structure* expresses organisational norms and regulations related to roles. (4) The *communicative structure* specifies the ontologies for description of domain concepts and communication illocutions.

2.2 Coordination Level

The *coordination level* provides a means to specify the patterns of interaction between services, and to transform the organisational representation coming from the organisational level into service-oriented workflows. The Coordination Design Tool can be used

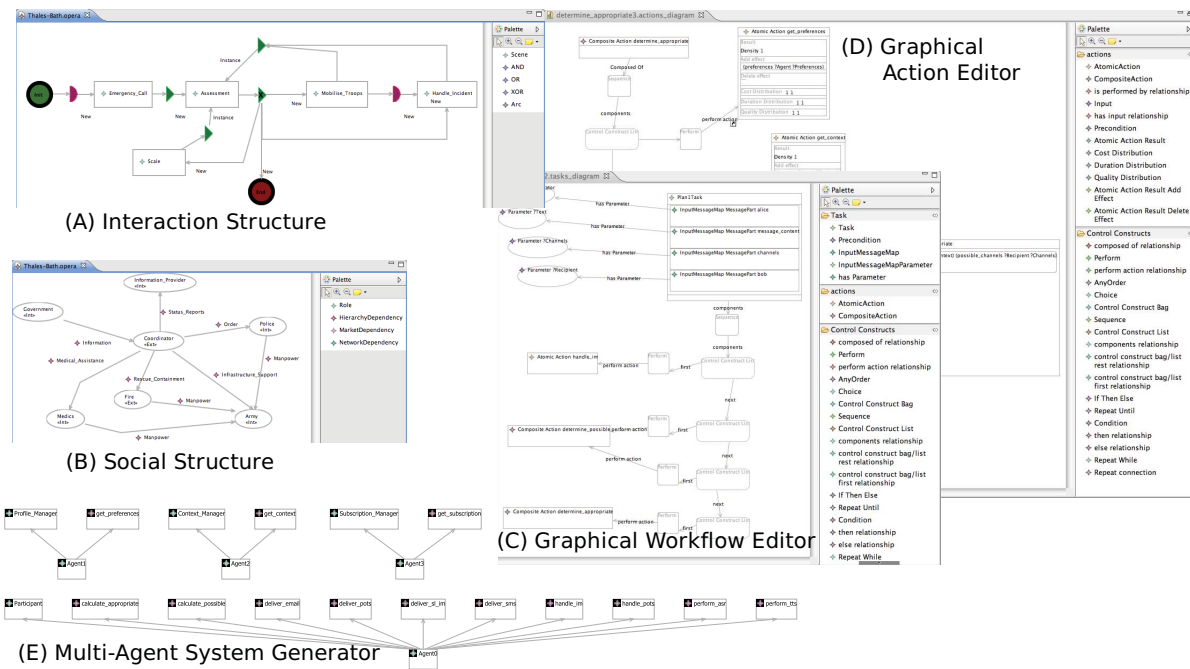


Figure 1: Screen shots from Eclipse: (A) & (B) Interaction and Social Structure Editor; (C) Graphical Workflow Editor; (D) Graphical Action Editor; (E) Multi-agent System

to design the whole coordination level of a distributed system by means of actors, tasks, workflows and workflow coordination mechanisms. The tools also support the generation of the agents that will perform the actual coordination tasks and the inspection of predefined and generated workflows. The Graphical Action editor (see Fig. 1 (D)) produces machine processable action descriptions, which can be used by other components, such as the workflow synthesis and agent tab. A workflow is composed of a sequence of steps, where each step is associated with an action and its inputs, along with a link to the next step in the workflow. Generated workflows are initially used by the workflow editor (see Fig. 1 (C)), which supports inspection and edition of workflow and uploading/downloading workflows to/from the workflow repository. Workflows are subsequently used by the agents for enactment. The multi-agent generator (see Fig. 1 (E)) takes the OperettA model and the actions defined for the organisation and creates an initial multi-agent system where every role in the organisation is assigned to an agent and the actions are distributed to the agents according to the role. The Planning agent chooses a plan and sends it to all the coordination level agents; each agent enacts the actions it is responsible for. The agents run in the AgentScape² platform.

2.3 Service Level

The *service level* supports the semantic description of services and the selection of the most appropriate service for a given task. It also effectively supports higher level and dynamic service composition. The Service Design Tool is used to generate or inspect service descriptions, edit service templates and register them in the service directory. It also connects with the service matchmaking tool (a human interface to the matchmaker component), allowing administrators to search for services matching a given task description or implementing a given service template and registering it in the service directory. The Service Setup Tool is used to check and modify the setup of the running environment, including the URIs of different resources, facilitating components, pre-defined services

and service containers.

The Monitor Tool allows administrators to inspect the status of a system's execution, and to keep track of the events generated at execution time and inspect how the system handles them. The tool aggregates and analyses event logs related to the execution of services, the fulfilment of coordination plans and the achievements of role and/or organisational objectives; and hence feedback is provided to the organisational model and workflow generation.

3. Why is it innovative?

The distinctiveness of our approach lies in the incorporation of an organisational context (mainly organisational objectives, structure and regulations) that can be used to dynamically select, compose and invoke services, providing an organisational awareness to some components (such as Agentified services at the Coordination Level or Matchmakers at the Service Level) that can direct the system execution in order to maintain the focus on higher-level organisational objectives. One of the effects is that exceptions can be managed not only at the lower level (as in other SOA architectures) but at higher levels, looking for alternative ways to fulfill a task or even a full abstract workflow by agreeing a new plan of action. Furthermore, organisational and coordination models offer a level of abstraction that allows humans, especially end-users, to better support the design and maintenance of the system.

Additional authors

S. Alvarez-Napagao and I. Gomez (Univ. Polit cnica de Catalunya), J. SC. Lam and D. Corsar (University of Aberdeen), H. Aldewereld and V. Dignum (Universiteit Utrecht), M. De Vos, D. Andreou and O. Cliffe (University of Bath), A. Staikopoulos (Trinity College Dublin), P. Sergeant and C. Reed (Calico Jack Ltd), T. Quillinan and K. Nieuwenhuis (Thales Nederland B.V.).

²<http://www.agentscape.org/>